

Math 5040

Topic 1: Simulation

Prof. S. Ethier

Reference: Sheldon M. Ross, *Introduction to Probability Models*, Chapter 11.
On 3-hour reserve in the Marriott Library.

<http://www.math.utah.edu/~ethier/5040.html>

The strong law of large numbers tells us: If X_1, X_2, \dots is an i.i.d. sequence with specified distribution and with X_1 having finite mean μ , then

$$\lim_{n \rightarrow \infty} \frac{X_1 + \dots + X_n}{n} = \mu \quad \text{a.s.} \quad (1)$$

Therefore, if we cannot evaluate μ directly, we may be able to estimate it via

$$\mu \approx \frac{X_1 + \dots + X_n}{n}, \quad (2)$$

provided we can generate realizations of the random variables X_1, \dots, X_n for some large n . This is called *Monte Carlo simulation* and dates back to the dawn of the computer age (1946).

Ordinarily, we generate uniform(0, 1) random variables and then create the specified random variables from them in some way. To generate a uniform(0, 1) random variable, we call a *random number generator*, which typically work like this: Given positive integers a , c , and m , and a *seed* X_0 , we define recursively

$$X_{n+1} \equiv (aX_n + c) \pmod{m}, \quad n \geq 0. \quad (3)$$

Then $X_1/m, X_2/m, \dots, X_n/m$ should be approximately an i.i.d. uniform(0, 1) sequence (at least for n much smaller than m). One popular choice is

$$a = 16807, \quad c = 0, \quad m = 2^{31} - 1 = 2147483647. \quad (4)$$

Notice that, if U is uniform(0, 1), then $a + (b - a)U$ is uniform(a, b).

Application to Buffon's needle problem: A uniform needle of length L is dropped on a table marked with parallel lines a distance $D \geq L$ apart. What is the probability that the needle crosses a line?

The usual solution is to let X be the distance between the midpoint of the needle to the nearest line, and let Θ be the acute angle between the needle and the projected line of length X . Then X and Θ are independent and uniform on $(0, D/2)$ and $(0, \pi/2)$, respectively. Hence

$$\begin{aligned} P(\text{needle crosses}) &= P((L/2) \cos \Theta > X) \\ &= \frac{4}{\pi D} \int_0^{\pi/2} \int_0^{(L/2) \cos \theta} dx d\theta = \frac{2L}{\pi D}. \quad (5) \end{aligned}$$

True BASIC program

```
RANDOMIZE
LET needles=100000
LET D=1
LET L=1
LET count=0
FOR n = 1 to needles
    LET X=(D/2)*rnd
    LET Th=(pi/2)*rnd
    IF (L/2)*cos(Th)>X then LET count = count + 1
NEXT n
PRINT using " .#####": 2/pi
PRINT using " .#####": count/needles
END
```

C++ program

```
#include <iostream> #include <stdlib.h>
#include <math.h>
int main ()
{
int needles=10000000, count=0, n;
double x, d=1., l=1., th, ratio;
double pi=3.14159265358979323846;
for (n=1; n<=needles; n++)
    {
    x=(d/2.)*drand48 ();
    th=(pi/2.)*drand48 ();
    if ((l/2.)*cos(th)>x) count++;
    }
ratio=(double)count/needles;
printf ("%15.12f\n%15.12f\n", 2./pi, ratio);
}
```

Mathematica program

```
needles = 1000000; d = 1; l = 1; count = 0;  
Do[{x = (d/2) Random[Real],  
    th = (Pi/2) Random[Real],  
    If[(1/2) Cos[th] > x, count = count + 1]},  
    {n, 1, needles}];  
N[2/Pi]  
N[count/needles]
```

Notice that, if U is uniform(0, 1), then $\text{int}(mU) + 1$ is discrete uniform $\{1, 2, \dots, m\}$.

Algorithm for simulating a random permutation of length n :
Start with $1, 2, \dots, n$. Choose one of the n integers and place it at the end. Then choose one of the first $n - 1$ and place it in position $n - 1$ Then choose one of the first two and place it in position 2. Done.

Better way: Start with $1, 2, \dots, n$. Choose one of the n integers and exchange it with the one in position n . Then choose one of the first $n - 1$ and exchange it with the one in position $n - 1$ Then choose one of the first two and exchange it with the one in position 2. Done.

Here we formalize the last algorithm:

1. Start with $(x(1), x(2), \dots, x(n)) = (1, 2, \dots, n)$.
2. Set $k = n$.
3. Let U be a random number and set $j = \text{int}(kU) + 1$.
4. Interchange $x(j)$ with $x(k)$.
5. Let $k = k - 1$.
6. If $k \geq 2$ then go to step 3.
7. $(x(1), x(2), \dots, x(n))$ is a random permutation.

```
RANDOMIZE
LET n=52
DIM x(52)
FOR i=1 to n
    LET x(i)=i
NEXT i
FOR i=1 to n
    PRINT using " ###": x(i);
NEXT i
PRINT
PRINT
FOR k=n to 2 step -1
    LET j=int(k*rnd) + 1
    LET temp=x(j)
    LET x(j)=x(k)
    LET x(k)=temp
NEXT k
```

```
FOR i=1 to n
  PRINT using " ###": x(i);
NEXT i
END
```

Application: The game of rencontre (“coincidence”). Shuffle a deck of n distinct cards labeled $1, 2, \dots, n$ and turn them over one by one. What is the probability that, for some k ($1 \leq k \leq n$), card k will be the k th card revealed?

Answer:

$$\sum_{m=1}^n \frac{(-1)^{m-1}}{m!} \approx 1 - e^{-1} \approx 0.632121. \quad (6)$$

```
LET n = 52
LET sample=10000
DIM x(52)
FOR i = 1 to n
    LET x(i) = i
NEXT i
LET total=0
RANDOMIZE
FOR run=1 to sample
    FOR k = n to 2 step -1
        LET I = int(k*rnd) + 1
        LET temp = x(I)
        LET x(I) = x(k)
        LET x(k) = temp
    NEXT k
    LET co=0
```

```
FOR k=1 to n
    IF x(k)=k then LET co=1
NEXT k
LET total=total+co
NEXT run
PRINT "sample size";
PRINT using "#####": sample
PRINT "exact    ";
PRINT using " .#####": 1-exp(-1)
PRINT "estimate";
PRINT using " .#####": total/sample
END
```

Another application: Find the probability of a *refait* of 31 at *trente et quarante* (“thirty and forty”). Six decks (312 cards) are mixed together. Aces count 1, picture cards count 10, all others count their nominal value. Deal cards until the total is 31 or greater. Repeat the process with the remaining deck. What is the probability that the two totals are both equal to 31?

Solution: Simulate a random permutation of 312 cards. Card k has value

$$\min(10, \text{int}((k - 1)/24)) + 1). \quad (7)$$

Notice at most 44 cards are needed, so we don't need to shuffle the entire deck.

```
RANDOMIZE
DIM x(312)
LET sample=10000
LET count=0
FOR run=1 to sample
  FOR i=1 to 312
    LET x(i)=i
  NEXT i
  FOR k=312 to 269 step -1
    LET j=int(k*rnd)+1
    LET temp=x(j)
    LET x(j)=x(k)
    LET x(k)=temp
  NEXT k
  LET sum1=0
  LET k=312
```

```
DO until sum1>=31
    LET sum1=sum1+min(10,int((x(k)-1)/24)+1)
    LET k=k-1
LOOP
LET sum2=0
DO until sum2>=31
    LET sum2=sum2+min(10,int((x(k)-1)/24)+1)
    LET k=k-1
LOOP
IF sum1=31 and sum2=31 then LET count=count+1
NEXT run
PRINT "exact 0.021891370"
PRINT count/sample
END
```

Inverse transformation method

First, we consider continuous random variables. Let X have distribution function $F(x) = P(X \leq x)$. The X can be simulated by generating a random number U and calculating $F^{-1}(U)$.

Why? Because, letting $X = F^{-1}(U)$, we have

$$P(X \leq x) = P(F^{-1}(U) \leq x) = P(U \leq F(x)) = F(x). \quad (8)$$

Example: Simulate an exponential(λ) random variable X . Here $F(x) = 1 - e^{-\lambda x}$, so $F(x) = u$ implies $1 - e^{-\lambda x} = u$ or $1 - u = e^{-\lambda x}$ or $-\ln(1 - u) = \lambda x$ or $x = -\lambda^{-1} \ln(1 - u)$. Thus, $F^{-1}(u) = -\lambda^{-1} \ln(1 - u)$. Substituting U for u and noting that $1 - U$ has the same distribution as U , we conclude that

$$X = \frac{-\ln U}{\lambda} \text{ is exponential}(\lambda). \quad (9)$$

Inverse transformation method, discrete case

Now consider a discrete random variable. Let $P(X = x_j) = p_j$, where the x_j are distinct and $\sum p_j = 1$. Then let

$$\begin{aligned} X = x_1 & \text{ if } 0 < U \leq p_1, \\ X = x_2 & \text{ if } p_1 < U \leq p_1 + p_2, \\ X = x_3 & \text{ if } p_1 + p_2 < U \leq p_1 + p_2 + p_3, \\ & \vdots \\ X = x_j & \text{ if } \sum_{i=1}^{j-1} p_i < U \leq \sum_{i=1}^j p_i, \\ & \vdots \end{aligned}$$

Clearly, this defines a random variable with the specified distribution.

Example: If X is discrete uniform on $\{1, 2, \dots, m\}$, then the method yields $X = \text{int}(mU) + 1$.

Example: If X is geometric(p), then $X = j \geq 1$ iff

$$\sum_1^{j-1} (1-p)^{i-1} p < U \leq \sum_1^j (1-p)^{i-1} p \quad (10)$$

iff

$$1 - (1-p)^{j-1} < U \leq 1 - (1-p)^j \quad (11)$$

iff

$$j \ln(1-p) \leq \ln(1-U) < (j-1) \ln(1-p) \quad (12)$$

iff

$$j-1 < \frac{\ln(1-U)}{\ln(1-p)} \leq j \quad (13)$$

Conclude that

$$X = \left\lfloor \frac{\ln U}{\ln(1-p)} \right\rfloor + 1. \quad (14)$$

The rejection method

Suppose we can simulate a random variable Y with density $g(y)$ and we want to simulate a random variable X with density $f(x)$. We assume only that

$$\frac{f(y)}{g(y)} \leq c \quad \text{for all } y \text{ with } g(y) > 0. \quad (15)$$

- Algorithm:
1. Simulate Y with density $g(y)$ and simulate an independent random number U .
 2. If $U \leq f(Y)/(cg(Y))$, then set $X = Y$. Otherwise return to step 1.
 3. X has density $f(x)$.

The last assertion requires proof.

Proof of the rejection method

$$\begin{aligned}P(X \leq x) &= P\{Y \leq x \mid U \leq f(Y)/(cg(Y))\} \\&= P\{Y \leq x, U \leq f(Y)/(cg(Y))\}/K \\&= \int P(Y \leq x, U \leq f(Y)/(cg(Y)) \mid Y = y) g(y) dy / K \\&= \int_{-\infty}^x [f(y)/(cg(y))] g(y) dy / K \\&= \int_{-\infty}^x f(y) dy / (cK),\end{aligned}\tag{16}$$

where $K = P\{U \leq f(Y)/(cg(Y))\}$. Letting $x \rightarrow \infty$, this gives $1 = 1/(cK)$, so $K = 1/c$ as required.

```
! Simulation of  $f(x)=L*x^a*(1-x)^b$ 
LET a=7
LET b=3
DIM freq(0:99), edf(0:99), fac(0:50)
LET fac(0)=1
FOR n=1 to 50
    LET fac(n)=n*fac(n-1)
NEXT n
LET L=fac(a+b+1)/(fac(a)*fac(b))
LET r=a/(a+b)
LET c=L*r^a*(1-r)^b
RANDOMIZE
LET M=100000
```

```
FOR n=1 to M
  LET U=1
  LET Y=0
  DO until U<=L*Y^a*(1-Y)^b/c
    LET Y=rnd
    LET U=rnd
  LOOP
  LET X=Y
  LET X0=int(100*X)
  LET freq(X0)=freq(X0)+1
NEXT n
FOR k=0 to 99
  FOR j=0 to k
    LET edf(k)=edf(k)+freq(j)
  NEXT j
NEXT k
```

```

OPEN #1: screen 0, 1, 0, 1
FOR n=0 to 99
    PLOT LINES: (n+1/2)/100,edf(n)/M;
                (n+3/2)/100,edf(n)/M
NEXT n
FOR n=0 to 999
    LET x=n/1000
    LET y=L*x^(a+1)/(a+1)
    FOR k=1 to b
        LET y=y+L*(-1)^k
            *(fac(b)/(fac(k)*fac(b-k)))
            *x^(a+k+1)/(a+k+1)
    NEXT k
    PLOT POINTS: x,y
NEXT n
END

```

In this last example, we took (with a, b positive integers)

$$f(x) = Lx^a(1-x)^b, \quad g(x) = 1, \quad 0 \leq x \leq 1. \quad (17)$$

By properties of the beta distribution, we know that

$$L = \frac{\Gamma(a+1)\Gamma(b+1)}{\Gamma(a+b+1)} = \frac{(a+b)!}{a!b!}. \quad (18)$$

We needed to maximize $f(x)/g(x) = f(x)$ over $0 \leq x \leq 1$.

Solving $f'(x) = 0$ we got $r = a/(a+b)$, hence the required max is $c = Lr^a(1-r)^b$. Finally, we needed the distribution function corresponding to the density $f(x)$, and this is

$$\begin{aligned} F(x) &= \int_0^x f(y) dy = \int_0^x Ly^a(1-y)^b dy \\ &= L \int_0^x \sum_{k=0}^b \binom{b}{k} (-1)^k y^{a+k} dy \\ &= L \sum_{k=0}^b \binom{b}{k} (-1)^k x^{a+k+1} / (a+k+1). \end{aligned} \quad (19)$$

In the last example, we found that $c = 2.93510725$, which means we can expect 1.935 rejections before each acceptance. In other words, c is the mean number of times we apply the rejection method for each simulated value obtained. That is because c is the mean of a geometric random variable with parameter $K = 1/c$.

The rejection method, discrete case

Suppose we can simulate a discrete random variable Y with mass function $P(Y = i) = q_i$ and we want to simulate a random variable X with the same range and mass function $P(X = i) = p_i$. We assume only that

$$\frac{p_i}{q_i} \leq c \quad \text{for all } i \text{ with } q_i > 0. \quad (20)$$

- Algorithm:
1. Simulate Y with mass function $P(Y = i) = q_i$ and simulate an independent random number U .
 2. If $U \leq p_Y / (c q_Y)$, then set $X = Y$. Otherwise return to step 1.
 3. X has density $P(X = i) = p_i$.

The last assertion requires proof.

Proof of the rejection method, discrete case

It is essentially an application of Bayes's theorem:

$$\begin{aligned}P(X = i) &= P\{Y = i \mid U \leq p_Y/(cq_Y)\} \\&= P\{Y = i, U \leq p_Y/(cq_Y)\}/K \\&= P(U \leq p_Y/(cq_Y) \mid Y = i) P(Y = i)/K \\&= [p_i/(cq_i)] q_i/K \\&= p_i/(cK),\end{aligned}\tag{21}$$

where $K = P\{U \leq p_Y/(cq_Y)\}$. Summing over all i , this gives $1 = 1/(cK)$, so $K = 1/c$ as required.

Example

Suppose we want to simulate a $\text{Poisson}(\lambda)$ random variable, so $f(n) = e^{-\lambda} \lambda^n / n!$ for each $n \geq 0$. We know how to simulate a $\text{geometric}(p)$ random variable less 1, so $g(n) = (1 - p)^n p$ for each $n \geq 0$. We need to maximize

$$\frac{f(n)}{g(n)} = \frac{\lambda^n e^{-\lambda} / n!}{(1 - p)^n p} = \frac{(\lambda / (1 - p))^n e^{-\lambda}}{n! p}, \quad (22)$$

which has the form of a $\text{Poisson}(\lambda / (1 - p))$ distribution except for a multiplicative constant. Now $f(n + 1) / f(n) = \lambda / (n + 1)$, so $f(n)$ is maximized at $n = \lfloor \lambda \rfloor$. Thus, $f(n) / g(n)$ is maximized at $n = \lfloor \lambda / (1 - p) \rfloor$.

Note: There are better ways to simulate a Poisson r.v.

```
! Simulation of Poisson by rejection method
LET lambda=3
LET p=1/6
DIM freq(0:20), fac(0:100)
LET fac(0)=1
FOR n=1 to 100
    LET fac(n)=n*fac(n-1)
NEXT n
LET m=int(lambda/(1-p))
LET c=((lambda/(1-p))^m/fac(m))*exp(-lambda)/p
PRINT c
RANDOMIZE
LET M=100000
```

```

FOR n=1 to M
  LET U=1
  LET Y=0
  DO until U<=((lambda/(1-p))^Y/fac(Y))
    *exp(-lambda)/(c*p)
    LET Y=int(log(rnd)/log(1-p))
    LET U=rnd
  LOOP
  LET X=Y
  LET freq(X)=freq(X)+1
NEXT n
FOR n=0 to 20
  PRINT using "###":n;
  PRINT using " .#####":freq(n)/M,
    exp(-lambda)*lambda^n/fac(n)
NEXT n
END

```

How many iterations are needed to apply the rejection method?
In both the continuous and the discrete cases, the distribution of the number of iterations is geometric with parameter K , hence mean $1/K = c$. It must necessarily be the case that $c \geq 1$, but we want to choose c as close to 1 as possible, to minimize the time required.

In the beta example, we found that $c = 2.93511$.

In the Poisson example, we found that $c = 2.32287$.

If X_1, X_2, \dots is an i.i.d. sequence with specified distribution and with X_1 having mean μ and variance $\sigma^2 > 0$, and if $\bar{X}_n = (X_1 + \dots + X_n)/n$, then

$$\lim_{n \rightarrow \infty} \bar{X}_n = \mu \quad \text{a.s.} \quad (23)$$

and

$$\frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}} \rightarrow N(0, 1) \quad \text{as } n \rightarrow \infty. \quad (24)$$

Thus, we can use S_n/n as an estimate of μ , and its standard deviation is σ/\sqrt{n} . To estimate the latter, we can use

$$\text{standard error} = \sqrt{\frac{1}{n} \sum_{i=1}^n X_i^2 - \bar{X}_n^2} / \sqrt{n}. \quad (25)$$

One should always report the standard error along with the simulated value.

Determining sample size. Suppose we want our simulation estimate to be accurate to within ε with probability about $1 - \alpha$, that is,

$$P(|\bar{X}_n - \mu| \leq \varepsilon) \approx 1 - \alpha. \quad (26)$$

Then we want $\varepsilon/(\sigma/\sqrt{n}) \approx z_{1-\alpha/2}$ or

$$n \approx \left(\frac{z_{1-\alpha/2}\sigma}{\varepsilon} \right)^2. \quad (27)$$

Here σ must be estimated, perhaps using a preliminary run.

Example: What is the dealer's advantage in single-deck blackjack, to within 0.001 with 95% confidence, if the player mimics the dealer?

Background: We assume that the game is dealt from a single standard 52-card deck. Aces have value 1 or 11 as specified below, picture cards (J, Q, K) have value 10, and every other card has value equal to its nominal value. Suits are irrelevant. The dealer receives two cards initially (one face up) and additional cards one at a time as needed to achieve a total of 17 or greater. The first ace has value 11 unless that would result in a total greater than 21, in which case it has value 1. Every subsequent ace has value 1. A total that includes an ace valued as 11 is called a *soft* total; every other total is called a *hard* total. E.g., $(8, 8, 7) = 23$, $(A, 6) = \text{soft } 17$, $(A, 5, 6, 7) = \text{hard } 19$.

More background: Problem calls for expected player loss. Player's hand is dealt first. Player loses 1 if player busts (total over 21) or if dealer's total exceeds player's and is at most 21. Player wins $3/2$ if player has 2-card 21 and dealer does not. Player wins 1 if dealer busts and player does not or if player's total exceeds the dealer's total and is at most 21. Otherwise, the hand results in a tie. Notice that there are two departures from complete symmetry.

To simulate, need to shuffle enough cards to deal two hands (15 cards suffice). Then determine player's and dealer's totals and hence player's loss ($1, 0, -1, -3/2$). Keep running total as sim is repeated. First, we'll write the sim, then we'll estimate σ , and this will allow us to determine the necessary sample size for the desired precision.

```
DIM x(52), y(15)
FOR i = 1 to 52
    LET x(i) = i
NEXT i
RANDOMIZE
LET sample=100000
LET cumprofit=0
let cumprofit2=0
FOR run=1 to sample

    ! shuffle cards and deal out first 15
    FOR k = 52 to 38 step -1
        LET j = int(k*rnd) + 1
        LET temp = x(j)
        LET x(j) = x(k)
        LET x(k) = temp
    NEXT k
```

```
FOR i = 1 to 15
    LET y(i)=min(10,int((x(53-i)-1)/4)+1)
NEXT i

! determine player's total
LET player=0
LET psoft=0
LET pcard=1
LET pbj=0
DO until player>=17 or psoft=1 and player>=7
    and player<=11
    LET player=player+y(pcard)
    IF y(pcard)=1 then LET psoft=1
    LET pcard=pcard+1
LOOP
IF player<=11 then LET player=player+10
IF player=21 and pcard=3 then LET pbj=1
```

```
! determine dealer's total
LET dealer=0
LET dsoft=0
LET dcard=pcard
LET dbj=0
DO until dealer>=17 or dsoft=1 and dealer>=7
                                and dealer<=11
    LET dealer=dealer+y(dcard)
    IF y(dcard)=1 then LET dsoft=1
    LET dcard=dcard+1
LOOP
IF dealer<=11 then LET dealer=dealer+10
IF dealer=21 and dcard=pcard+2 then LET dbj=1

! determine dealers's profit
IF dbj=1 and pbj=0 then LET profit=1
IF pbj=1 and dbj=0 then LET profit=-1.5
```

```
IF pbj=1 and dbj=1 then LET profit=0

IF dbj=0 and player>21 then LET profit=1
IF dbj=0 and dealer<=21 and player<dealer
    then LET profit=1
IF pbj=0 and player<=21 and dealer<player
    then LET profit=-1
IF pbj=0 and player<=21 and dealer>21
    then LET profit=-1
IF pbj=0 and dbj=0 and player<=21 and
    player=dealer then LET profit=0
LET cumprofit=cumprofit+profit
Let cumprofit2=cumprofit2+profit^2
NEXT run
PRINT cumprofit/sample,
    cumprofit2/sample-(cumprofit/sample)^2
END
```

Example: The program produced the numbers .056705 and .961392.

The required sample size for accuracy within .001 with probability .95 is

$$n \approx \left(\frac{z_{1-\alpha/2}\sigma}{\epsilon} \right)^2 \approx \frac{1.96^2(0.961392)}{0.001^2} = 3,693,000. \quad (28)$$

Now rerun the program with this sample size. For simplicity, we took $n = 4,000,000$ and the program took about 13 minutes and produced the numbers .0572624 and .959985. The former number is indeed within .001 of the exact probability, which is about .056846. Curiously, our simulation based on 100,000 runs was more accurate, but that is just a coincidence.